MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/17

ECE2216 MICROCONTROLLER AND MICROPROCESSOR SYSTEMS

(All sections / Groups)

13 OCTOBER 2016
9:00 a.m. – 11:00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 10 pages including cover page and appendix with 4 Questions only.

2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please print all your answers in the Answer Booklet provided.

Question 1

(a) Given two signed numbers $4B_{16}$, and 24_8:

i. Convert them into 7-bit binary format. [2 marks]

ii. Add these two numbers obtained from Q1(a)(i). [1 mark]

iii. Identify the result obtained from Q1(a)(ii) as a positive or a negative number. Comment on the result obtained. [2 marks]

Hint: Consider the numbers in Q1(a)(i) are already in 2's complement representation.

(b) Illustrate a diagram to show the connections between the Central Processing Unit (CPU), Input-Output (I/O) and internal memory of a microcontroller. [4 marks]

(c) Highlight TWO features in term of Instruction Set perspective to distinguish microcontrollers from microprocessors. [2 marks]

(d) A memory structure is shown in Figure Q1(d) for the 8051 microcontroller.

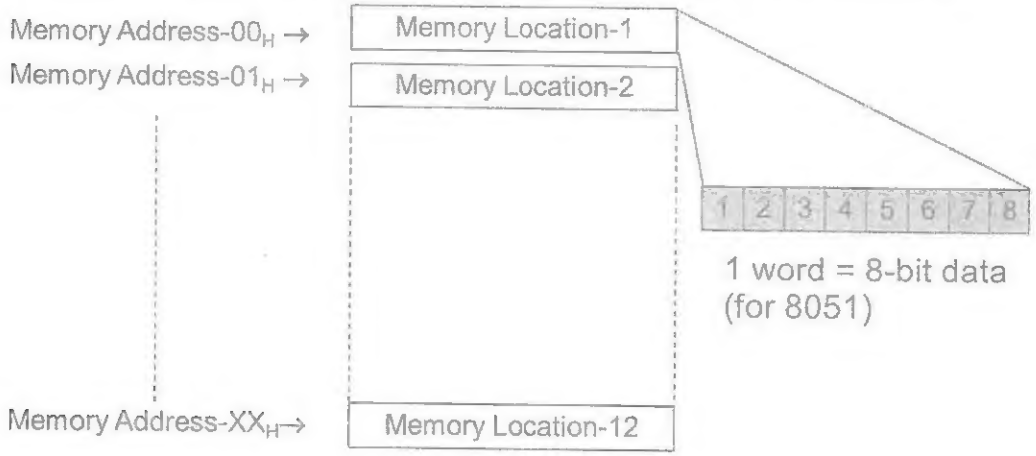

Figure Q1(d)

i. What is the size of the data bus? [1 mark]

ii. What is the size of the address bus for the memory structure shown in Figure Q1(d)? [3 marks]

iii. Find the value for memory address XX_H. [2 marks]

(e) Answer TRUE or FALSE for the following questions:

i. Intel 8086 is a 16-bit microprocessor.

ii. Multicore processor uses slightly more power than the multi-processor implementation.

iii. An octa-core processor contains eight processing cores.

[3 marks]

Continued ...

(f) The pin configuration of an 8051 microcontroller is shown in Figure Q1(f).

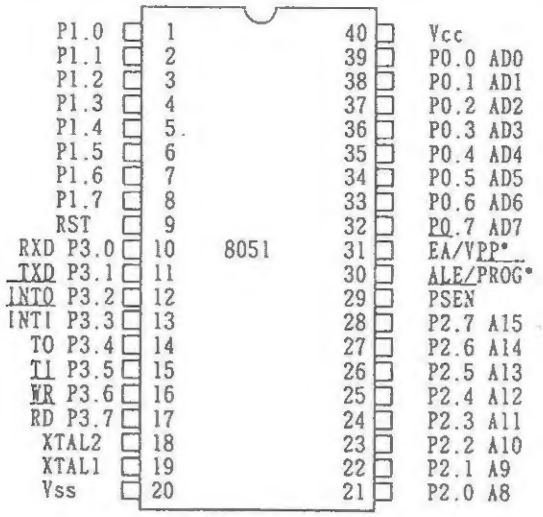

Figure Q1(f)

List all pins to be used in accessing an external 4 Kbyte ROM. [5 marks]

Continued ...

Question 2

(a) An 8051 microcontroller is required to be configured with 32 Kbytes of external ROM and 32 Kbytes of external RAM. Assume each RAM and ROM memory blocks are available as 32 Kbytes. Draw the configuration of the system showing the 8051 signal lines to be used for address, data and control buses. [11 marks]

(b) Figure Q2(b) below shows an 8051 microcontroller fetching a byte of instruction (10H) from the external ROM at memory location (1234H).

i. What are the values of "X", "Y" and "Z" in Figure Q2(b)? [3 marks]

ii. Explain the purpose of having $\overline{\text{PSEN}}$ control signal. [3 marks]

iii. If the 8051 microcontroller proceeds to read a byte of data from the external RAM, what is the control signal involved instead of $\overline{\text{PSEN}}$? [1 mark]

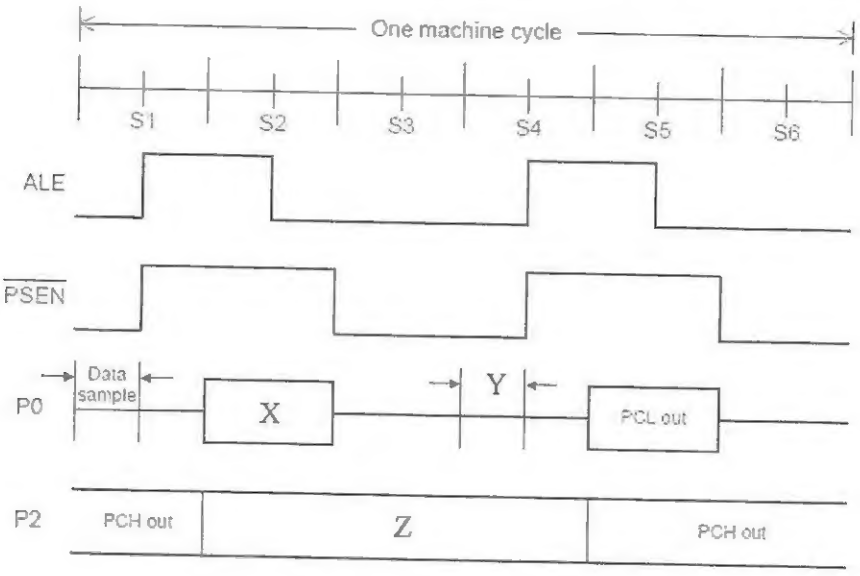

Figure Q2(b)

(c) A simple arithmetic MCS-51 instruction sequence is shown in Figure Q2(c).

```
        MOV R0, #3
        MOV R1, #30H
        MOV A, #00000001B
REPEAT: MOV B, #3
        MUL AB
        DJNZ R0, REPEAT
        MOV @R1, A
```

Figure Q2(c)

i. Specify the addressing mode used in the source and destination operands of "MOV @R1, A". [2 marks]

ii. Explain the function of "DJNZ R0, REPEAT". [3 marks]

iii. What are the final contents (in hexadecimal) of accumulator A and register B? [2 marks]

Continued...

Question 3

(a) An MCS-51 assembly instruction sequence is shown in Figure Q3(a).

```
MOV     A, #5AH
SWAP    A
ADD     A, #5AH
```

Figure Q3(a)

i. Convert the assembly instructions to the corresponding machine codes. [3 marks]

ii. Determine the execution time of this instruction sequence if it is executed on an 8051 microcontroller with a 12 MHz crystal oscillator. [2 marks]

iii. What is the content of accumulator A after all three instructions have been executed? [2 marks]

Hint: Refer to the opcode map in the Appendix.

(b) An assembly program for the 8051 microcontroller has been written as shown in Figure Q3(b):

```
        ORG 0000H
        MOV R0, #30H
        MOV R1, #2
        MOV R2, #0
        MOV R3, #3
        MOV DPTR, #TBL
LOOP:   MOV A, R1
        MOVC A, @A+DPTR
        MOV @R0, A
        INC R0
        INC R1
        ADD A, R2
        MOV R2, A
        DJNZ R3, LOOP
        SJMP $          ; the program stops here
TBL:    DB 01H, 04H, 09H, 10H, 19H, 24H
        END
```

Figure Q3(b)

i. Analyze the program listed in Figure Q3(b) to determine the address of the affected memory locations in the RAM and their final contents when the program has stopped. [6 marks]

ii. Determine the final contents of registers R0, R1, R2 and accumulator A when the program has stopped. [6 marks]

(c) Develop an assembly program for the 8051 microcontroller to generate a 500Hz square wave (50% duty cycle) at pin P1.0 by using Timer 0 operating in Mode 1. Assume that the microcontroller is operating with a 12 MHz crystal oscillator. [6 marks]

Continued...

Question 4

(a) The 8051 microcontroller is equipped with an on-chip serial port for communication with external devices using UART.

i. The instruction sequence listed in Figure Q4(a) is used to initialized the on-chip serial port. Determine the mode of operation for the serial port and the baud rate that is setup by this instruction sequence. Assume that the 8051 microcontroller is connected to an 11.0592 MHz crystal oscillator [4 marks]

```
MOV   SCON, #52H
MOV   TMOD, #21H
MOV   TH1, #0FAH
ORL   PCON, #10000000B
SETB  TR1
```

Figure Q4(a)

ii. If the 8051 microcontroller has been initialized by the instruction sequence listed in Figure Q4(a), write a subroutine called "SEND_YES" to transmit the string "YES" serially. [4 marks]

(b) As a trainee in the embedded system engineering department, you are required to carry out the following tasks for an 8051 microcontroller operating with a 12 MHz crystal oscillator.

i. Write an instruction sequence to enable External 0, Timer 0 and Timer 1 interrupts, and set Timer 1 interrupt to have the highest priority followed by External 0 interrupt and Timer 0 interrupt. [3 marks]

ii. Write an instructions sequence to setup Timer 1 in Mode 2 with appropriate reload value to produce a 2 kHz square wave (50% duty cycle). [3 marks]

iii. Write the interrupt service routine for Timer 1 interrupt to toggle pin P0.0 each time Timer 1 overflows. [2 marks]

(c) Three common cathode seven-segment displays are connected to an 8051 microcontroller as shown in Figure Q4(c). Develop an assembly program for the 8051 microcontroller to display '1' on Digit 2, '2' on Digit 1 and '3' on Digit 0. [9 marks]

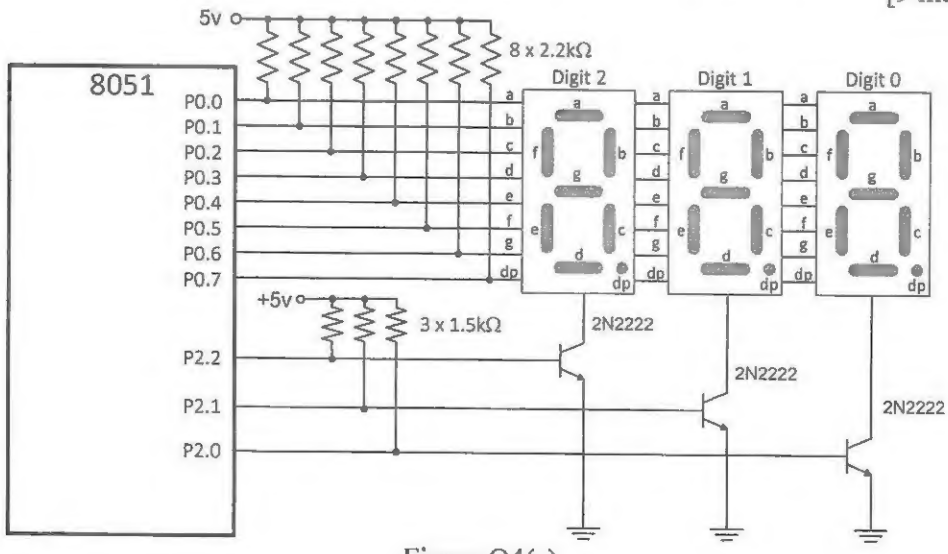

Figure Q4(c)

End of Paper

APPENDIX

Special Function Register Formats

Interrupt Enable (IE)

Bit Addr.	AFH	-	-	ACH	ABH	AAH	A9H	A8H
Name	**EA**	-	-	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IE.7	EA	Global enable/disable. EA = 1, each individual source is enabled/disabled by setting/clearing its enable bit. EA = 0, disable all interrupts.
IE.6	-	Undefined
IE.5	-	Not implemented in 8051. ET2 for 8052.
IE.4	ES	Serial port interrupt enable bit.
IE.3	ET1	Timer 1 interrupt enable bit.
IE.2	EX1	External interrupt enable bit.
IE.1	ET0	Timer 0 interrupt enable bit.
IE.0	EX0	External interrupt enable bit.

Interrupt Priority (IP)

Bit Addr.	-	-	-	BCH	BBH	BAH	B9H	B8H
Name	-	-	-	**PS**	**PT1**	**PX1**	**PT0**	**PX0**

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IP.7	-	Undefined
IP.6	-	Undefined
IP.5	-	Not implemented in 8051. PT2 for 8052.
IP.4	PS	Serial port interrupt priority bit.
IP.3	PT1	Timer 1 interrupt priority bit.
IP.2	PX1	External interrupt priority bit.
IP.1	PT0	Timer 0 interrupt priority bit.
IP.0	PX0	External interrupt priority bit.

Interrupt Vectors

Interrupt Source	Flag	Vector Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI & TI	0023H
Timer 2 (8052)	TF2 or EXF2	002BH

Program Status Word (PSW)

Bit Addr.	D7H	D6H	D5H	D4H	D3H	D2H	-	D0H
Name	**CY**	**AC**	**F0**	**RS1**	**RS0**	**0V**	-	**P**

Serial Control (SCON)

Bit Addr.	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
Name	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

BIT	**SYMBOL**	**FUNCTION**
SCON.7	SM0	Serial port mode bit 0 (see Table A.1).
SCON.6	SM1	Serial port mode bit 1 (see Table A.1).
SCON.5	SM2	Serial port mode bit 2; enables multiprocessor communications in modes 2 and 3; RI will not be activated if received 9th bit is 0. In mode 1, if SM2 = 1, then RI will be activated only if a valid stop bit was received. In mode 0, SM2 should be 0.
SCON.4	REN	Receiver enable; must be set to receive characters.
SCON.3	TB8	Transmit bit 8; 9th bit transmitted in modes 2 and 3; set/cleared by software.
SCON.2	RB8	Receive bit 8; 9th bit received.
SCON.1	TI	Transmit interrupt flag; set at end of character transmission; cleared by software.
SCON.0	RI	Receive interrupt flag; set at end of character reception; cleared by software.

Table A.1 The 8051 Serial Port Mode Selection

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift register	Fixed
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fixed
1	1	3	9-bit UART	Variable

PCON (Power Control)

Bit	7	6	5	4	3	2	1	0
Name	**SMOD**	-	-	-	**GF1**	**GF0**	**PD**	**IDL**

BIT	**SYMBOL**	**FUNCTION**
7	SMOD	Double the serial port baud rate when it is set to 1
3	GF1	General purpose flag bit 1
2	GF0	General purpose flag bit 0
1	PD	Activate the microcontroller power-down mode when it is set to 1
0	IDL	Activate the microcontroller idle mode when it is set to 1

Timer Control (TCON)

Bit Addr.	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
Name	**TF1**	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer-1 overflow flag. Set by hardware on overflow. Cleared by hardware when processor vectors to interrupt routine. Must be cleared by software when not involve interrupt
TCON.6	TR1	Timer-1 run control bit. Set/cleared by software to turn timer/counter on/off.
TCON.5	TF0	Timer-0 overflow flag. Do the same function as TF1 but for Timer-0.
TCON.4	TR0	Timer-0 run control bit. Do the same function as TR1 but for Timer-0.
TCON.3	IE1	External interrupt-1 edge flag. Set by hardware when interrupt-1 falling edge is detected. Cleared by hardware when interrupt is processed.
TCON.2	IT1	Interrupt-1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	External interrupt-1 edge flag. Do the same function as IE1 but for external interrupt-0.
TCON.0	IT0	Interrupt-0 Type control bit. Do the same function as IT1 but for external interrupt-0.

Timer Mode (TMOD)

Bit	7	6	5	4	3	2	1	0
Name	**GATE**	**C/T**	**M1**	**M0**	**GATE**	**C/T**	**M1**	**M0**

BIT	SYMBOL	FUNCTION
TMOD.7	GATE1	When this bit is set the timer will only run when INT1 (P3.3) is high (hardware control). When this bit is cleared the timer will run regardless of the state of INT1 (software control).
TMOD.6	C/T1	Timer / Counter select bit. C / $\overline{T}$ = 0 → Timer operation. C / $\overline{T}$ = 1 → Counter operation.
TMOD.5	M1	Mode selection bits (see Table A.2). [for timer 1]
TMOD.4	M0	Mode selection bits (see Table A.2). [for timer 1]
TMOD.3	GATE0	Exactly the same function as GATE1 but for Timer0
TMOD.2	C/T0	Exactly the same function as C/T1 but for Timer0
TMOD.1	M1	Mode selection bits (see Table A.2). [for timer 0]
TMOD.0	M0	Mode selection bits (see Table A.2). [for timer 0]

Table A.2 Timer Mode Selection

M1	M0	Timer Mode	Description of Mode
0	0	0	13-bit Timer
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

Opcode Map

byte / Instruction operands / cycle	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1B NOP 1C	3B JBC bit, rel 2C	3B JB bit, rel 2C	3B JNB bit, rel 2C	2B JC rel 2C	2B JNC rel 2C	2B JZ rel 2C	2B JNZ rel 2C	2B SJMP rel 2C	3B MOV DPTR, #data16 2C	2B ORL C, /bit 2C	2B ANL C, /bit 2C	2B PUSH dir 2C	2B POP dir 2C	1B MOVX A, @DPTR 2C	1B MOVX @DPTR, A 2C
1	2B AJMP (P0) 2C	2B ACALL (P0) 2C	2B AJMP (P1) 2C	2B ACALL (P1) 2C	2B AJMP (P2) 2C	2B ACALL (P2) 2C	2B AJMP (P3) 2C	2B ACALL (P3) 2C	2B AJMP (P4) 2C	2B ACALL (P4) 2C	2B AJMP (P5) 2C	2B ACALL (P5) 2C	2B AJMP (P6) 2C	2B ACALL (P6) 2C	2B AJMP (P7) 2C	2B ACALL (P7) 2C
2	3B LJMP addr16 2C	3B LCALL addr16 2C	1B RET 2C	2B RETI 2C	2B ORL dir, A 1C	2B ANL dir, A 1C	2B XRL dir, A 1C	2B ORL C, bit 2C	2B ANL C, bit 2C	2B MOV bit, C 2C	2B MOV C, bit 1C	2B CPL bit 1C	2B CLR bit 1C	2B SETB bit 1C	1B MOVX A, @R0 2C	1B MOVX @R0, A 2C
3	1B RR A 1C	1B RRC A 1C	1B RL A 1C	1B RLC A 1C	3B ORL dir, #data 2C	3B ANL dir, #data 2C	3B XRL dir, #data 2C	1B JMP @A+DPTR 2C	1B MOVC A, @A+PC 2C	1B MOVC A, @A+DPTR 2C	1B INC DPTR 2C	1B CPL C 1C	1B CLR C 1C	1B SETB C 1C	1B MOVX A, @R1 2C	1B MOVX @R1, A 2C
4	1B INC A 1C	1B DEC A 1C	2B ADD A, #data 1C	2B ADDC A, #data 1C	2B ORL A, #data 1C	2B ANL A, #data 1C	2B XRL A, #data 1C	2B MOV A, #data 1C	1B DIV AB 4C	2B SUBB A, #data 1C	1B MUL AB 4C	3B CJNE A, #data, rel 2C	1B SWAP A 1C	1B DA A 1C	1B CLR A 1C	1B CPL A 1C
5	2B INC dir 1C	2B DEC dir 1C	2B ADD A, dir 1C	2B ADDC A, dir 1C	2B ORL A, dir 1C	2B ANL A, dir 1C	2B XRL A, dir 1C	3B MOV dir, #data 2C	3B MOV dir, dir 2C	2B SUBB A, dir 1C		3B CJNE A, dir, rel 2C	2B XCH A, dir 1C	3B DJNZ dir, rel 2C	2B MOV A, dir 1C	2B MOV dir, A 1C
6	1B INC @R0 1C	1B DEC @R0 1C	1B ADD A, @R0 1C	1B ADDC A, @R0 1C	1B ORL A, @R0 1C	1B ANL A, @R0 1C	1B XRL A, @R0 1C	2B MOV @R0, #data 1C	2B MOV dir, @R0 2C	1B SUBB A, @R0 1C	2B MOV @R0, dir 2C	3B CJNE @R0, #data,rel 2C	1B XCH A, @R0 1C	1B XCHD A, @R0 1C	1B MOV A, @R0 1C	1B MOV @R0, A 1C
7	1B INC @R1 1C	1B DEC @R1 1C	1B ADD A, @R1 1C	1B ADDC A, @R1 1C	1B ORL A, @R1 1C	1B ANL A, @R1 1C	1B XRL A, @R1 1C	2B MOV @R1, #data 1C	2B MOV dir, @R1 2C	1B SUBB A, @R1 1C	2B MOV @R1, dir 2C	3B CJNE @R1, #data,rel 2C	1B XCH A, @R1 1C	1B XCHD A, @R1 1C	1B MOV A, @R1 1C	1B MOV @R1, A 1C
8	1B INC R0 1C	1B DEC R0 1C	1B ADD A, R0 1C	1B ADDC A, R0 1C	1B ORL A, R0 1C	1B ANL A, R0 1C	1B XRL A, R0 1C	2B MOV R0, #data 1C	2B MOV dir, R0 2C	1B SUBB A, R0 1C	2B MOV R0, dir 2C	3B CJNE R0, #data, rel 2C	1B XCH A, R0 1C	2B DJNZ R0, rel 2C	1B MOV A, R0 1C	1B MOV R0, A 1C
9	1B INC R1 1C	1B DEC R1 1C	1B ADD A, R1 1C	1B ADDC A, R1 1C	1B ORL A, R1 1C	1B ANL A, R1 1C	1B XRL A, R1 1C	2B MOV R1, #data 1C	2B MOV dir, R1 2C	1B SUBB A, R1 1C	2B MOV R1, dir 2C	3B CJNE R1, #data, rel 2C	1B XCH A, R1 1C	2B DJNZ R1, rel 2C	1B MOV A, R1 1C	1B MOV R1, A 1C
A	1B INC R2 1C	1B DEC R2 1C	1B ADD A, R2 1C	1B ADDC A, R2 1C	1B ORL A, R2 1C	1B ANL A, R2 1C	1B XRL A, R2 1C	2B MOV R2, #data 1C	2B MOV dir, R2 2C	1B SUBB A, R2 1C	2B MOV R2, dir 2C	3B CJNE R2, #data, rel 2C	1B XCH A, R2 1C	2B DJNZ R2, rel 2C	1B MOV A, R2 1C	1B MOV R2, A 1C
B	1B INC R3 1C	1B DEC R3 1C	1B ADD A, R3 1C	1B ADDC A, R3 1C	1B ORL A, R3 1C	1B ANL A, R3 1C	1B XRL A, R3 1C	2B MOV R3, #data 1C	2B MOV dir, R3 2C	1B SUBB A, R3 1C	2B MOV R3, dir 2C	3B CJNE R3, #data, rel 2C	1B XCH A, R3 1C	2B DJNZ R3, rel 2C	1B MOV A, R3 1C	1B MOV R3, A 1C
C	1B INC R4 1C	1B DEC R4 1C	1B ADD A, R4 1C	1B ADDC A, R4 1C	1B ORL A, R4 1C	1B ANL A, R4 1C	1B XRL A, R4 1C	2B MOV R4, #data 1C	2B MOV dir, R4 2C	1B SUBB A, R4 1C	2B MOV R4, dir 2C	3B CJNE R4, #data, rel 2C	1B XCH A, R4 1C	2B DJNZ R4, rel 2C	1B MOV A, R4 1C	1B MOV R4, A 1C
D	1B INC R5 1C	1B DEC R5 1C	1B ADD A, R5 1C	1B ADDC A, R5 1C	1B ORL A, R5 1C	1B ANL A, R5 1C	1B XRL A, R5 1C	2B MOV R5, #data 1C	2B MOV dir, R5 2C	1B SUBB A, R5 1C	2B MOV R5, dir 2C	3B CJNE R5, #data, rel 2C	1B XCH A, R5 1C	2B DJNZ R5, rel 2C	1B MOV A, R5 1C	1B MOV R5, A 1C
E	1B INC R6 1C	1B DEC R6 1C	1B ADD A, R6 1C	1B ADDC A, R6 1C	1B ORL A, R6 1C	1B ANL A, R6 1C	1B XRL A, R6 1C	2B MOV R6, #data 1C	2B MOV dir, R6 2C	1B SUBB A, R6 1C	2B MOV R6, dir 2C	3B CJNE R6, #data, rel 2C	1B XCH A, R6 1C	2B DJNZ R6, rel 2C	1B MOV A, R6 1C	1B MOV R6, A 1C
F	1B INC R7 1C	1B DEC R7 1C	1B ADD A, R7 1C	1B ADDC A, R7 1C	1B ORL A, R7 1C	1B ANL A, R7 1C	1B XRL A, R7 1C	2B MOV R7, #data 1C	2B MOV dir, R7 2C	1B SUBB A, R7 1C	2B MOV R7, dir 2C	3B CJNE R7, #data, rel 2C	1B XCH A, R7 1C	2B DJNZ R7, rel 2C	1B MOV A, R7 1C	1B MOV R7, A 1C